

# CLASSIFICATION OF CRESCENT CONFIGURATIONS

REBECCA F. DURST, MAX HLAVACEK, CHI HUYNH, STEVEN J. MILLER, AND EYVINDUR A. PALSSON

**ABSTRACT.** Let  $n$  points be in crescent configurations in  $\mathbb{R}^d$  if they lie in general position in  $\mathbb{R}^d$  and determine  $n - 1$  distinct distances, such that for every  $1 \leq i \leq n - 1$  there is a distance that occurs exactly  $i$  times. Since Erdős' conjecture in 1989 on the existence of  $N$  sufficiently large such that no crescent configurations exist on  $N$  or more points, he, Pomerance, and Palásti have given constructions for  $n$  up to 8 but nothing is yet known for  $n \geq 9$ . Most recently, Burt et. al. [BGMMPs] had proven that a crescent configuration on  $n$  points exists in  $\mathbb{R}^{n-2}$  for  $n \geq 3$ . In this paper, we study the classification of these configurations on 4 and 5 points through graph isomorphism and rigidity. Our techniques, which can be generalized to higher dimensions, offer a new viewpoint on the problem through the lens of distance geometry and provide a systematic way to construct crescent configurations.

## CONTENTS

1. Introduction	2
2. Classification of Crescent Configurations	3
3. Method for Counting Isomorphism Classes	6
4. Geometrically Realizability of Crescent Configurations	9
5. Finding Geometric Realizations for Crescent Configurations	11
6. Rigidity of Crescent Configurations	11
6.1. Preliminaries	12
6.2. Rigidity Analysis for 4-point Crescent Configurations	13
7. Future Work	16
7.1. Further Explorations in the Plane	16
7.2. Extensions to Higher Dimensions	16
7.3. Properties of Crescent Configuration Types	16
Appendix A.	17
Appendix B.	19
References	22

---

*Date:* October 26, 2016.

2010 *Mathematics Subject Classification.* 52C10 (primary) 52C35 (secondary).

This work was supported financially by Williams College Finnerty Fund, Simons Foundation Grant #360560, NSF Grants DMS1265673, DMS1561945 and NSF Grant DMS1347804. We want to thank Williams College and SMALL REU for continuous support, Kevin Kwan for helpful conversations on geometrical properties of crescent configurations and Ileana Streinu for insightful advice on a better proof of rigidity.

## 1. INTRODUCTION

Erdős once wrote, “my most striking contribution to geometry is, no doubt, my problem on the number of distinct distances,” [Erd96]. The referred question, which asks what is the minimum number of distinct distances determined by  $n$  points, was first asked in 1946 [Erd46] and marked the beginning of a chain of variants. See [She] and [BGMMP] for a survey on these. Although one would expect all distances between  $n$  points to be different if they were to be placed in the plane at random, if the distances are regularly placed, such as on a lattice, then many distances may repeat. Erdős’ conjectured lower bound,  $\Omega(n/\sqrt{\log n})$ , attained by a  $\sqrt{n} \times \sqrt{n}$  integer lattice, was essentially proven up to a  $\sqrt{\log n}$  factor by Guth and Katz in 2010. [GK]

The variant we study in this paper is one where the distances have prescribed multiplicities. One says  $n$  points are in crescent configuration in  $\mathbb{R}^d$  if they are in general position and determine  $n - 1$  distinct distances such that for every  $1 \leq i \leq n - 1$ , there is a distance that occurs exactly  $i$  times. Erdős conjectured that there exists a sufficiently large  $N$  such that no crescent configuration exists on  $N$  or more points [Erd89]. Though constructions have been provided for  $n = 5, 6, 7, 8$  by Erdős, I. Palásti and C. Pomerance [Pal87, Pal89, Erd89], little progress has been made towards a construction for  $n \geq 9$ . One problem often encountered in the search for these configurations is the lack of understanding of their properties and the difficulty in exhibiting the configurations’ information combinatorially.

As such, we take a new approach to studying these crescent configurations, borrowing techniques from distance geometry and graph theory. Our main theorems are the results of two algorithms that search for and classify all crescent configurations on any  $n \geq 4$  up to graph isomorphism and find geometric realizations for each of these isomorphism classes in the plane.

**Theorem 1.1.** *Given a set of three distinct distances  $\{d_1, d_2, d_3\}$  on four points, there are only three allowable crescent configurations up to graph isomorphism. In Figure 1 we provide graph realizations for each type.*

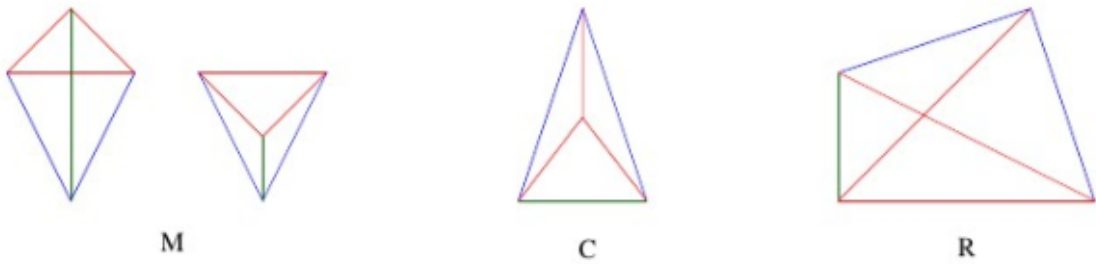


FIGURE 1. Types M, C, and R.

**Theorem 1.2.** *Given a set of four distinct distances  $\{d_1, d_2, d_3, d_4\}$  on five points, there are 27 allowable crescent configurations up to graph isomorphism. In Figure 2 we provide graph realizations for each type.*

The advantage of this algorithmic method is that it can be applied in higher dimensions, though we hope that the running time ( $\mathcal{O}(n^n)$ ) can still be vastly improved.

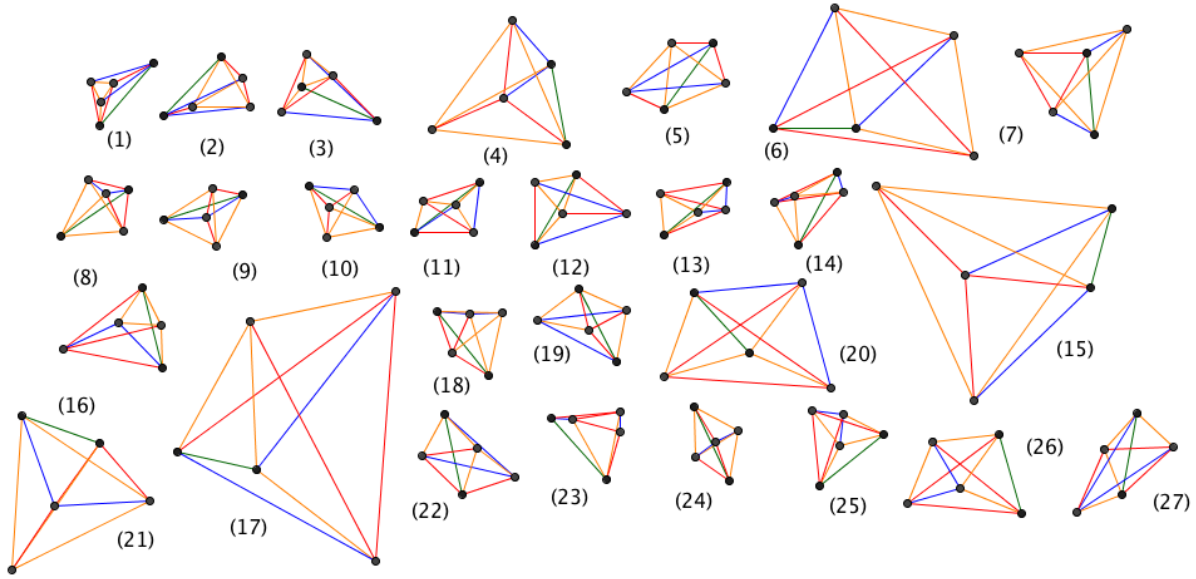


FIGURE 2. Representatives for all possible crescent configurations on five points.

In Appendix B we include distance sets and realizable distances for each crescent configuration on four and five points.

In Section 2, we introduce our distance geometry approach and prove a classification of crescent configurations for a general  $n$ . We follow this in Section 3 with an outline of the first half of the algorithm used to achieve Theorems 1.1 and 1.2. We then move to Section 4 where we discuss how distance geometry methods may be applied to determine whether a distance set is realizable. In Section 5, we outline the second half of the algorithm, completing the proofs for Theorems 1.1 and 1.2. After the theorems have been established, we discuss another way to verify the uniqueness of our isomorphism classes and also a different method to characterize these configurations through analysis of rigidity in Section 6. Lastly, we discuss potential future work based on our approach using graph theory, rigidity, and distance geometry.

**Remark 1.3.** *The authors are happy to provide copies of any code referenced in the course of this paper. Please email [Steven.Miller.MC.96@aya.yale.edu](mailto:Steven.Miller.MC.96@aya.yale.edu).*

## 2. CLASSIFICATION OF CRESCENT CONFIGURATIONS

In this section we provide the key definitions and theorems that we use to classify crescent configurations.

**Definition 2.1** (General Position [BGMMPs]). *We say that  $n$  points are in general position in  $\mathbb{R}^d$  if no  $d+1$  points lie on the same hyperplane and no  $d+2$  lie on the same hypersphere.*

**Definition 2.2** (Crescent Configuration [BGMMPs]). *We say  $n$  points are in crescent configuration (in  $\mathbb{R}^d$ ) if they lie in general position in  $\mathbb{R}^d$  and determine  $n - 1$  distinct distances, such that for every  $1 \leq i \leq n - 1$  there is a distance that occurs exactly  $i$  times.*

The notion of general position is very important in the construction of crescent configuration. Without this notion, the problem of placing  $n$  points in  $\mathbb{R}^d$  to determine  $n - 1$  distinct distances

satisfying the prescribed multiplicities becomes trivial. By simply placing  $n$  points on a line in an arithmetic progression, we solve the problem in any dimension.

**Definition 2.3** (Distance Coordinate). *The distance coordinate,  $D_A$ , of a point  $A$  is the multiset of all distances, counting multiplicity, between  $A$  and the other points in a set  $\mathcal{P}$ . Order does not matter.*

**Definition 2.4** (Distance Set). *The distance set,  $\mathcal{D}$  corresponding to a set of points,  $\mathcal{P}$ , is the multiset of the distance coordinates for each point in the  $\mathcal{P}$ .*

**Definition 2.5** (Isomorphism for Labeled Graph [HHH]). *Graph  $A$  is isomorphic to graph  $B$  if and only if there exists a bijective function  $f : V(A) \mapsto V(B)$ , (where  $V(A)$  and  $V(B)$  are the vertex spaces) such that:*

- (1)  $\forall a_i \in A, l_A(a_i) = l_B(f(a_i))$ ,
- (2)  $\forall a_i, a_j \in V, \{a_i, a_j\} \in E_A \leftrightarrow \{f(a_i), f(a_j)\} \in E_B$ , and
- (3)  $\forall \{a_i, a_j\} \in E_A, w_A(\{a_i, a_j\}) = w_B(f(\{a_i, a_j\}))$ ,

where  $\{l_A, l_B\}$  and  $\{w_A, w_B\}$  are functions that define the labels of the vertices and edges of  $A$  and  $B$  respectively.

We note that a crescent configuration on  $n$  points can be considered a weighted complete graph with  $n - 1$  distinct weights associated to the edges in a certain manner, so that the configuration can be realized in  $\mathbb{R}^d$ . The adjacency matrix, thus, is a natural way to store information about the configuration. Should we rearrange the weights that each vertex is incident to, we would most likely have to draw a configuration differently. Due to this insight, we have the following theorem.

**Theorem 2.6.** *Let  $A$  and  $B$  be two crescent configurations on the same number of points  $n$ . If  $A$  and  $B$  have the same distance sets, then there exists a graph isomorphism from  $A$  to  $B$ .*

*Proof.* Consider two crescent configurations,  $A$  and  $B$ , each on  $n$  points,  $\{a_1, \dots, a_n\}$  and  $\{b_1, \dots, b_n\}$ , such that  $A$  and  $B$  have the same distance sets (up to order of the entries of the coordinates). We note that we do not yet care about the specific distances and, instead, designate each distance by the points that define them, as will be shown below.

First we show that the ordering of the elements of the distance coordinates does not matter when comparing the distance sets. Then we prove that  $A$  and  $B$  are isomorphic when we view them as labeled graphs.

Consider  $A$ . First, we number each point of  $A$  from 1 to  $n$ . For each point  $a_i$ ,  $1 \leq i \leq n$ , in the configuration, we re-order its distance coordinate so that the distance between  $a_i$  and  $a_1$ ,  $d_{a_i,1}$ , is in the first slot, the distance between  $a_i$  and  $a_2$ ,  $d_{a_i,2}$ , is in the second slot, and so on, inserting 0 into the  $i$ th slot, representing the distance between  $a_i$  and  $a_i$ . We will call these *augmented distance coordinates*. All of our augmented distance coordinates uniquely determine the points that they represent. Note that the arrangement of these augmented coordinates depends on how we choose to index the points of  $A$ . The nonzero entries, however, only encode information about the set of distances between a point and all other points in a given configuration. They do not account for relative position. That is to say, until assign a number to each of the points of  $A$  (i.e.,  $a_1, a_2$ , and so on), the order of the entries of the distance coordinates does not matter. The distance coordinate  $\{d_1, d_2, d_3\}$  is the same as the coordinate  $\{d_3, d_1, d_2\}$ .

Next we do the same thing for  $B$ . Since the distance sets of  $A$  and  $B$  are the same, we know that for every augmented coordinate in  $A$ , there is an augmented coordinate in  $B$  with the same set of non-zero entries. Thus, let  $a_i$  and  $b_j$  have the same non-zero entries in their augmented coordinates. We define each point by its augmented coordinate,

$$D_{a_i} = (d_{a_i,1}, \dots, d_{a_i,i-1}, 0, d_{a_i,i+1}, \dots, d_{a_i,n})$$

$$D_{b_j} = (d_{b_j,1}, \dots, d_{b_j,j-1}, 0, d_{b_j,j+1}, \dots, d_{b_j,n}).$$

Since the set of non-zero entries in  $D_{a_i}$  and  $D_{b_j}$  are the same, we know that there exists a permutation,  $f_{ij}$  such that  $f_{ij}(D_{a_i}) = D_{b_j}$ . Since we know that every distance coordinate in  $A$  has a match in  $B$ , there exists a permutation function of this sort that maps  $D_{a_i}$  to some  $D_{b_j}$  for all  $1 \leq i \leq n$ .

Furthermore, we actually only need one permutation for all of the reordered coordinates of  $A$ . This can be quickly seen by recognizing that the ordering of a single augmented distance coordinate of  $A$  determines the ordering of all augmented distance coordinates if we wish to retain all information. To prove that it is indeed the case, we point out that the reordered distance coordinates of  $A$  are, in fact, the rows and columns of a weighted adjacency matrix. The adjacency matrix below models a configuration on 3 points. The element in row 1, column 2 represents the distance between points  $a_1$  and  $a_2$ .

$$\begin{bmatrix} 0 & d_{1,2} & d_{1,3} \\ d_{1,2} & 0 & d_{2,3} \\ d_{1,3} & d_{2,3} & 0 \end{bmatrix}$$

From this, a proof by induction will show that if we permute the entries of row 1, the other rows (and columns) must undergo the same permutation in order to preserve the symmetry of the matrix.

We now identify each point of  $A$  and  $B$  by these augmented and rematched distance coordinates to create two labelled and weighted graphs where the vertices are the points of  $A$  and  $B$ , and each edge  $\{a_i, a_j\}$  is weighted by the distance between the two points,  $d_{a_i,j}$ . In these graphs, the vertex labels are the augmented distance coordinates of each point, where the coordinates of  $A$  are further transformed by  $f_{ij}$ , and the edge labels are the distinct distances,  $d_1, d_2, \dots, d_{n-1}$  that define our crescent configurations. Given this information, it is very straightforward to prove that these graphs are isomorphic by showing that they satisfy the three conditions in Definition 2.5.

Let  $F : V(A_D) \mapsto V(B_D)$  be such that  $F(a_i) = b_j$ , where  $b_j$  is chosen so that the distance coordinates of  $a_i$  and  $b_j$  have the same nonzero values. Since we augmented and transformed our distance coordinates by the permutation function  $f_{ij}$ , we know that  $a_i$  and  $b_j$  have the same vertex labels. Thus, we know  $F$  is one-to-one because the distance sets of  $A$  and  $B$  are the same. Furthermore, we know that  $F$  is onto because  $A$  and  $B$  both have  $n$  points. Therefore,  $\forall b_j \in B$ ,  $\exists a_i \in A$  such that  $F(D_{a_i}) = D_{b_j}$ . Therefore,  $F$  is a bijection.

*Condition 1:* Let  $l_A$  and  $l_B$  be defined as in Definition 2.5. This means that  $l_A(a_i) = f_{ij}(D_{a_i}) = D_{b_j}$  and  $l_B(b_j) = D_{b_j}$ . Thus, given  $F(a_i) = b_j$ , we have that  $\forall a_i \in A$ ,  $l_A(a_i) = l_B(F(a_i))$ .

*Condition 2:* This condition holds because crescent configurations are complete graphs, and  $F$  is a bijection. Therefore  $\{a_i, a_j\} \in E_A \leftrightarrow \{F(a_i), F(a_j)\} \in E_B$ , where  $E_A$  and  $E_B$  are the edge sets of  $A$  and  $B$  respectively.

*Condition 3:* To prove this condition, we recall that applying  $f_{ij}$  to  $a_i$  is equivalent to re-indexing the points of  $A$ . Therefore, if we let  $w_A$  and  $w_B$  be functions that return the edge labels of  $A$  and  $B$ , respectively, as defined as in Definition 2.5, then we know that

$$w_A(\{a_i, a_j\}) = d_{a_i,j} = d_{b_i,k} = w_B(\{b_i, b_k\}) = w_B(\{F(a_i), F(a_j)\}),$$

satisfying this condition. Thus  $A$  and  $B$  are isomorphic.  $\square$

**Remark 2.7.** *Since the algorithm can only distinguish the similar permutations according to the distance labels, the resulting distance sets may define crescent configurations that are not geometrically realizable. We address this concern later in the paper (see Section 4 on geometric realizability).*

### 3. METHOD FOR COUNTING ISOMORPHISM CLASSES

As a direct result of Theorem 2.6, we now have a method for classifying all crescent configurations on  $n$  points into isomorphism classes. In this section we provide a sketch of the algorithm used to find these isomorphism classes. A pseudocode is provided in Appendix A.

Consider a set of distances,  $\{d_1, d_2, d_2, \dots, d_{n-1}\}$ , associated to a crescent configuration on  $n$  points. This set of distances may be threaded through an  $n \times n$  adjacency matrix like the one shown in the proof of Theorem 2.6. From this point on, we refer to these matrices as *distance matrices*; however, they are the equivalent of weighted adjacency matrices in graph theory.

**Definition 3.1** (Distance Matrix). *Let  $A$  be an  $n \times n$  matrix. We say  $A$  is a **distance matrix** if the following conditions are satisfied*

- (1)  $A$  is symmetric,
- (2) the entries along the main diagonal of  $A$  are all 0, and
- (3)  $a_{ij}$  (the entry found in the  $i$ th row and  $j$ th column of  $A$ ) refers to the distance between points  $i$  and  $j$  of the point configuration (or graph) defined by  $A$ .

As each configuration has a distance matrix associated to it, we can generate all possible configurations by threading all permutations of  $\{d_1, d_2, d_2, \dots, d_{n-1}\}$  through distance matrices. Using standard combinatorial techniques, we can quickly see that this method will generate 60 configurations on four points, 12,600 on five points, and 37,837,800 on six points. We would now like to apply Theorem 2.6 and group these configurations into isomorphism classes (via graph isomorphism).

A computer program may then be used to group together all distance matrices defining configurations with identical distance sets. These groups then represent our isomorphism classes, and we can conduct the remainder of our analysis on one representative from each class. For  $n = 4$ , this reduces our initial 60 to 4 classes. For  $n = 5$ , it reduces our initial 12,600 to 85 classes.

Having finished this classification, we note that there are three degenerate cases that will force the configuration to *always* violate general position. These will remain unchanged under graph isomorphism since vertex and edge-set pairings stay the same, allowing us to eliminate entire isomorphism classes if one representative is proven to be degenerate. These cases are as follows.

- (1) The configuration contains one point at the center of a circle of radius  $d_i$  with four or more points on this circle as seen in Figure 3.
- (2) The configuration contains three (or more) isosceles triangles sharing the same base.
- (3) The configurations contains four points arranged on the vertices of an isosceles trapezoid.

Although there exist other cases that will force a class to violate general position, these three cases may be accounted for by only considering the distance matrices.

Case 1 is very simple to account for and is only possible for  $n \geq 5$ . In order to eliminate these cases, we remove configurations containing one or more distance coordinates in which a particular distance,  $d_i$ , occurs four or more times. In Algorithm A.1, this case and case 3 are accounted for in the procedure REMOVECYCLIC.

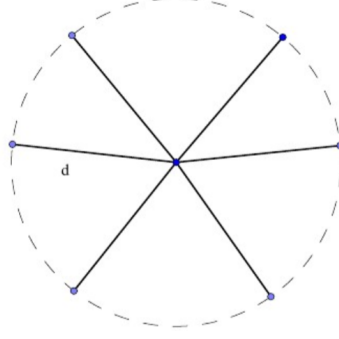


FIGURE 3. The central point of this configuration has distance coordinate  $\{d, d, d, d, d, d\}$ .

As with case 1, case 2 is only possible for  $n \geq 5$ . If three or more isosceles triangles share the same base, then all of their apexes must reside on the line bisecting this base, forcing them to violate general position.

In a distance matrix, an isosceles triangle is indicated by a matching pair of distances occurring in a row. Therefore, we remove all distance matrices in which three or more rows contain a matching pair occurring in the same slots in each row. This case is accounted for in Algorithm A.1 by the procedure REMOVELINEARCASE.

Case 3 requires us to remove all configurations that contain a subset or subsets of four points defining an isosceles trapezoid, since isosceles trapezoids are always cyclic quadrilaterals. In Algorithm A.1, the procedures SUBMATRICES and REMOVECYCLIC are included to account for these cases, which may be identified by their distance matrices using the following lemma.

**Lemma 3.2.** *A  $4 \times 4$  distance matrix will always define an isosceles trapezoid if and only if one of the following holds:*

- (1) *the matrix has only one distinct row such that*
  - (a) *the matrix has only two distinct distances, or*
  - (b) *the matrix only has three distinct distances,*
- (2) *the matrix has two distinct rows, both with multiplicity two, such that*
  - (a) *the matrix has three distinct distances with multiplicity no greater than three (note that this means each distance occurs no more than six times in the distance matrix), or*
  - (b) *the matrix has four distinct distances with multiplicity no greater than two (each distance occurs no more than four times in the distance matrix).*

*Proof.* ( $\Leftarrow$ ) According to Halsted [Hal], a necessary and sufficient condition for a quadrilateral to be an isosceles trapezoid is that it has at least one pair of opposite sides with equal length and diagonals of equal length. It is not possible for these two lengths (sides and diagonals) to be equal because this would create two isosceles triangles that would have to be congruent. Therefore there are three cases for isosceles trapezoids: (1) four distinct distances, (2) three distinct distances, or (3) two distinct distances. Figure 4 presents possible realizations for each of these cases. From here, it is straightforward to show that each of these quadrilaterals satisfies one of the conditions stated in Lemma 3.2, thus completing this direction of the proof.

( $\Rightarrow$ ) We now prove the other direction.

We begin with condition (1a): one distinct row and two distinct distances.

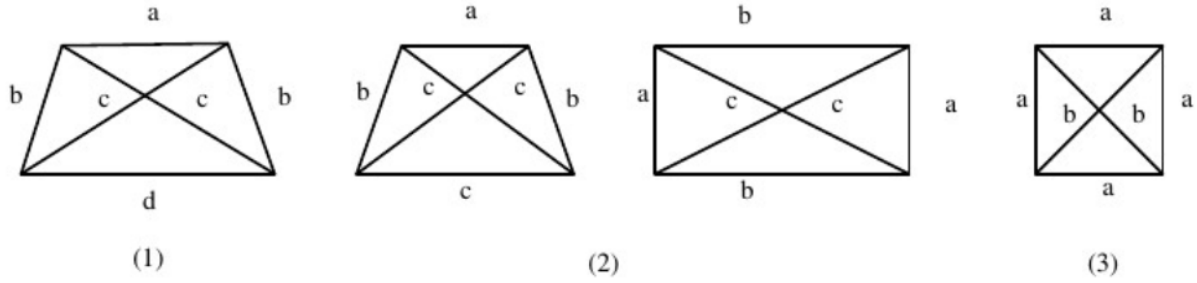


FIGURE 4. (1) Four distinct distances, (2) three distinct distances, (3) two distinct distances.

Assume each row represents the distance coordinate  $(a, a, b)$  (order of distances may differ among rows). Since we require all rows of the distance matrix to have the same distance coordinate, distance  $b$  must touch every point yet only show up twice. Therefore, it must represent both diagonals or two opposite sides. Both cases yield a quadrilateral with a set of congruent opposite sides and congruent diagonals, a necessary and sufficient condition for an isosceles trapezoid.

For condition 1b - one distinct row and three distinct distances - all three distances must be common to all points, so they must describe either a set of opposite sides or the diagonals, in all cases yielding an isosceles trapezoid.

The proofs in the other directions all following similar arguments. In each case there are two distances that must be common to all points and at least one distance that can only be common to two points, implying that one set of opposite sides cannot be congruent, therefore the distances common to all points must describe the diagonals and only one set of opposite sides, thus describing an isosceles trapezoid in each case.  $\square$

Once these cases have been eliminated, we are left with three isomorphism classes for four points and 51 for five points.

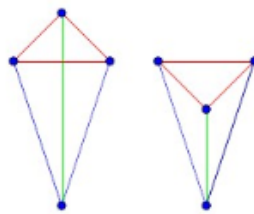


FIGURE 5. Two isomorphic configurations.

**Remark 3.3.** *It should be noted that the runtime of Algorithm A.1 is  $\mathcal{O}(n^n)$ , so its use is limited to crescent configurations on relatively few points. However, we believe that with enough processing power, upper bounds can be established using the algorithm for small  $n$  such as 7 and 8. As such, at this time, this does not pose much of an issue to the progress of this research, as no crescent configurations on more than 8 points is yet to be found.*



#### 4. GEOMETRICALLY REALIZABILITY OF CRESCENT CONFIGURATIONS

In the previous sections we developed a way to find every isomorphism class of distance sets that corresponded to a crescent configuration on  $n$  points; however, it is not clear that given some  $m$  and one of these distance sets, there exists a set of points in  $\mathbb{R}^m$  that realizes this distance set. Thus, we formally define geometric realizability.

**Definition 4.1.** *A crescent configuration on  $n$  points is geometrically realizable in  $\mathbb{R}^m$  if there exists some distances  $d_1, \dots, d_{n-1}$  for which there exist points  $P_1, P_2, \dots, P_n$  in  $\mathbb{R}^m$  that realize the corresponding distance set.*

We have thus far established upper bounds on the number of crescent configurations on  $n$  points based on the combinatorial aspects of this problem. We now sharpen these bounds by considering criteria for geometric realizability.

**Remark 4.2.** *Burt et. al. [BGMMPs] showed that given an  $n$ , there exists an  $m$  such that an  $n$ -crescent configuration exists in  $\mathbb{R}^m$ . In this section, we fix  $m$  and determine whether a given distance set is geometrically realizable in  $\mathbb{R}^m$ .*

In Section 3, we classify crescent configurations in terms of the distances between each pair of points. The problem of determining information about a set of points based on the distances between them is well studied, and is known as the distance geometry problem. Thus, we are now able to use techniques from distance geometry in order to sharpen the bounds on the number of geometrically realizable configurations on  $n$  points. We begin by introducing some of these techniques.

**Definition 4.3.** *The Cayley-Menger matrix corresponding to a set of  $n$  points  $\{P_1, P_2, \dots, P_n\}$  is an  $(n+1) \times (n+1)$  matrix of the following form:*

$$\begin{pmatrix} 0 & d_{1,2}^2 & \dots & d_{1,n}^2 & 1 \\ d_{2,1}^2 & 0 & \dots & d_{2,n}^2 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ d_{n,1}^2 & d_{n,2}^2 & \dots & 0 & 1 \\ 1 & 1 & \dots & 1 & 0 \end{pmatrix},$$

where  $d_{i,j}$  is the distance between  $P_i$  and  $P_j$ .

The Cayley-Menger matrix corresponding to a set of points can be used to determine whether those points lie in a  $d$ -dimensional Euclidian space. Note that  $n+1$  points always lie in a hyperplane of  $\mathbb{R}^n$ ; thus we only need to consider collections of points of size at least  $n+2$ .

**Theorem 4.4** (Cayley-Menger Matrix). [LL] *Let  $M$  be a  $(n+3) \times (n+3)$  matrix of the form specified above.  $M$  is the Cayley-Menger matrix of a set of  $n+2$  points in  $\mathbb{R}^n$  if and only if  $\det M = 0$ .*

**Corollary 4.5.** *Let  $M$  be a  $n+1 \times n+1$  matrix of the form specified above;  $M$  is the Cayley-Menger matrix of a set of  $n$  points,  $\{P_1, \dots, P_n\}$  in  $\mathbb{R}^2$ , if and only if the Cayley-Menger matrix corresponding to every size  $m+2$  subset has determinant zero.*

*Proof.*  $\Rightarrow$  Suppose that  $M$  is the Cayley-Menger matrix of  $n$  points that can be realized in  $\mathbb{R}^m$ . Thus, every subset of  $m+2$  points can also be realized in  $\mathbb{R}^m$ , and thus has a corresponding Cayley-Menger matrix with determinant zero.

$\Leftarrow$  Suppose that the Cayley-Menger matrix corresponding to every size  $m + 2$  subset has determinant zero. Note that  $P_1, \dots, P_{m+1}$  must define an  $m$ -dimensional subspace of any Euclidean space. Note that by assumption, for every  $i$ , the Cayley-Menger matrix corresponding to  $P_1, \dots, P_{m+1}, P_i$  has determinant 0. Thus, as a consequence of Theorem 4.4,  $P_i$  is in the same  $m$ -dimensional subspace. Thus, our set of  $n$  points must be realizable in  $\mathbb{R}^m$ .  $\square$

Submatrices of the Cayley-Menger matrix can also determine whether the points determined by a set of distances lie on the same hypersphere.

**Definition 4.6.** *The Euclidean distance matrix corresponding to a set of  $n$  points  $\{P_1, P_2, \dots, P_n\}$  is an  $n \times n$  matrix of the following form:*

$$\begin{pmatrix} 0 & d_{1,2}^2 & \dots & d_{1,n}^2 \\ d_{2,1}^2 & 0 & \dots & d_{2,n}^2 \\ \vdots & \vdots & \ddots & \vdots \\ d_{n,1}^2 & d_{n,2}^2 & \dots & 0 \end{pmatrix}.$$

Note that the Euclidean distance matrices defined above are slightly different than the general notion of distance matrices used in previous sections.

**Theorem 4.7.** [DS] *Let  $E$  be the Euclidean distance matrix corresponding to points  $P_1, \dots, P_n$  in Euclidean space. These points lie on a hypersphere in  $\mathbb{R}^{n-2}$  if and only if  $\det E = 0$ .*

We apply these techniques to the problem of counting crescent configurations in the following corollary.

**Corollary 4.8.** *Let  $M$  be the Cayley-Menger matrix corresponding to a distance set on  $n$  points,  $\mathcal{D}$ . Then  $\mathcal{D}$  is geometrically realizable in general position in  $\mathbb{R}^m$  if and only if the following conditions hold.*

- (1) *Let  $S$  be a size  $m + 3$  subset of  $\{1, 2, \dots, n, n + 1\}$  that contains  $n + 1$ . Let  $M_S$  be the submatrix of  $M$  with rows and columns indexed by  $S$ . For every choice of  $S$ ,  $\det M_S = 0$ .*
- (2) *Let  $S$  be a size  $m + 2$  subset of  $\{1, \dots, n, n + 1\}$ . For every such choice of  $S$ ,  $\det M_S \neq 0$ .*

*Proof.* From Corollary 4.5, a collection of distances between  $n$  points is geometrically realizable in  $\mathbb{R}^m$  if and only if the Cayley-Menger matrix corresponding to each subset  $m + 2$  points has determinant 0. Each of these matrices is one of the submatrices specified by the first condition.

We consider the submatrices specified by the second condition in two parts. First, consider the submatrices  $M_S$  for which  $n + 1 \in S$ . We see that these comprise the Cayley-Menger matrices for each subset of  $m + 1$  points. We now consider the submatrices  $M_S$  for which  $n + 1 \notin S$ . We see that these comprise the Euclidean distance matrices for each subset of  $m + 2$  points. Thus, we can see that the submatrices specified by the second condition is the set of all Euclidean distance matrices for each size  $m + 2$  subset of the  $n$  points together with the set of all Cayley-Menger matrices for each size  $m + 1$  subset of the  $n$  points. By Theorem 4.7, a Euclidean distance matrix on  $m + 2$  points has determinant 0 if and only if these points lie on the same hypersphere in  $\mathbb{R}^m$ . By Theorem 4.4, the Cayley-Menger matrix of  $m + 1$  points has determinant 0 if and only if these points lie on the same hyperplane in  $\mathbb{R}^m$ . Thus, this second condition holds if and only if the points lie in general position in  $\mathbb{R}^m$ .  $\square$

Our application of this corollary to the distance sets on 4 and 5 points have allowed us to determine the geometric realizability of each of the distance sets found using techniques from earlier sections. These geometrically realizable configurations are discussed in the following section.

Thus far, most of our attentions have been focused on crescent configurations in the plane. However, these techniques can be applied to finding crescent configurations in higher dimensions, furthering the work of Burt et. al. [BGMMPs].

## 5. FINDING GEOMETRIC REALIZATIONS FOR CRESCENT CONFIGURATIONS

As stated in Section 3, Algorithm A.1 yields three crescent configurations on four points and 51 on five points, unique up to isomorphism. However, these procedures do not guarantee that each isomorphism class contains geometrically realizable configurations.

To check which of these configurations are geometrically realizable, we run Algorithm A.3. A pseudocode of this algorithm can be found in Appendix A. Note that we assume  $d_1 = 1$  in order to simplify the procedure.

Algorithm A.3 is an extended application of Theorem 4.8. The first step of this algorithm is to take the Cayley-Menger determinants of all 4-point subsets of each configuration found by Algorithm A.1 and set them equal to zero. Doing so yields a system of  $\binom{n}{4}$  equations with unknowns :  $\{d_2, d_3, \dots, d_{n-1}\}$ . If the configuration is realizable in the plane, solving this system of equations will give all possible solutions to these distances in  $\mathbb{R}^2$ . Note that the values must be positive and real-valued.

For each of these solutions, we check the Cayley-Menger determinants of all 3-point subsets of the configuration. If one or more of these determinants equals zero, that solution forces the configuration to place three points on the same line, violating general position, so we throw it away. If none of the determinants are zero, we keep the solution.

For each remaining solution, we take the determinant of the Euclidean distance matrix of each 4-point subset of the configuration. If any of these determinants equal zero, the solution forces four points onto the same circle, violating general position, and we throw it away.

Any remaining solutions represent the distances of a geometrically realizable crescent configuration. Applying this algorithm to the configurations returned by Algorithm A.1 completes the proofs of Theorems 1.1 and 1.2, as we find that there are exactly three realizable crescent configurations on four points and 27 realizable crescent configurations on five points.

In Appendix B, we provide a set of distances for every configuration on five points that had at least one remaining solution after applying this algorithm.

## 6. RIGIDITY OF CRESCENT CONFIGURATIONS

So far in this paper, we have shown that crescent configurations can be classified into a finite number of isomorphism classes for each positive integer  $n$  and defined a method to geometrically realize these configurations. We now turn our attention to testing whether our crescent configurations are rigid, which can help answer questions, such as, whether one distance set could define two different realizations of crescent configurations belonging to the same isomorphism class. For this reason, we shall treat our configurations as graphs and adopt the following definition of graph rigidity.

**Definition 6.1** (Asimow & Roth). [AR] *Let  $G$  be a graph  $(V, E)$  on  $v$  vertices in  $\mathbb{R}^n$  then  $G(p)$  is  $G$  together with the point  $p = (p_1, p_2, \dots, p_v) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \dots \mathbb{R}^n = \mathbb{R}^{nv}$ . Let  $K$  be the*

complete graph on  $v$  vertices. The graph  $G(p)$  is rigid in  $\mathbb{R}^n$  if there exists a neighborhood  $\mathbf{U}$  of  $p$  such that

$$e_K^{-1}(e_K(p)) \cap \mathbf{U} = e_G^{-1}(e_G(p)) \cap \mathbf{U},$$

where  $e_K$  and  $e_G$  are the edge functions of  $K$  and  $G$ , which return the distances of edges of the associated graphs.

In other words, a rigid graph cannot have its vertices be continuously moved to noncongruent positions while preserving the distances. The rigidity testing of these configurations would not only serve as a verification for our classification under graph isomorphism, but would also give us another way to characterize these crescent configurations.

One important note is that all crescent configurations are complete graphs. It is a rather direct result of the following theorem and its corollary that all complete graphs are rigid in  $\mathbb{R}^2$ .

**Theorem 6.2** (Laman). [Lam] *The edges of a graph  $G = (V, E)$  are independent in two dimensions if and only if no subgraph  $G' = (V', E')$  has more than  $2n - 3$  edges.*

**Corollary 6.3.** *A graph with  $2n - 3$  edges is rigid in two dimensions if and only if no subgraph  $G'$  has more than  $2n' - 3$  edges.*

We can easily check that the complete graph on 1, 2 and 3 points ( $K_1$ ,  $K_2$  and  $K_3$  respectively) satisfy the condition and thus are rigid. For  $n \geq 3$ , the complete graph on  $n$  points (or  $K_n$  by convention) is composed of triangles, which are  $K_3$ . Since each  $K_3$  composing  $K_n$  is rigid, it must also be the case that one cannot move the vertices of  $K_n$  to noncongruent positions while preserving the distances because this would inevitably change the distances in each  $K_3$  subgraph. Therefore, all crescent configurations are rigid.

Nonetheless, this fact does not imply our work is over, as there is more than one kind of rigidity for graphs. Thus, we want to study whether all crescent configurations fall into just one category of rigidity or if there are certain crescent configurations that are more rigid than others. In addition, Theorem 6.2 can only be used to show the rigidity of crescent configurations in two dimensions. Furthermore, in the near future, we want to extend our definition of crescent configuration to  $\epsilon$  crescent configurations, where two edges are considered equal if their lengths are within a sufficiently small  $\epsilon$  from each other. Thus we need another method that could accommodate the exploration in higher dimensions and the assessment of stability of an extended family of crescent configurations.

This leads us to using **rigidity matrices**. These are an extremely powerful and flexible tool that can be used in  $\mathbb{R}^d$  for all  $d \in \mathbb{N}$ . We introduce them and other necessary terminologies in Subsection 6.1 and apply them to our analysis in Subsection 6.2.

**6.1. Preliminaries.** The following definitions and theorems will be used through out the rest of the paper to analyze the rigidity of crescent configurations on 4 and 5 points in  $\mathbb{R}^2$ . The formulation of each of these definitions have been adopted from one paper by Bruce Hendrickson [Hen] due to their accessible nature. For other formulations and characterizations of rigidity, see also [AR, Con, Rot80, Rot81].

**Definition 6.4** (Realization). [Hen] *Let  $G = (V, E)$  be a graph with some pairwise associated distance measurements. A realization  $f$  of  $G$  is a function that maps the vertices of  $G$  to coordinates in some Euclidean space such that the distance measurements are realized.*

**Definition 6.5** (Framework). [Hen] *A framework is a combination of a graph  $G = (V, E)$  and a realization of  $G$  in some Euclidean space, denoted  $f(G)$*

In our case, we will mostly be concerned with frameworks in  $\mathbb{R}^2$ , though the techniques can be generalized to higher dimension and in any Euclidean space.

**Definition 6.6** (Flexibility of Frameworks). [Hen] *A framework is called flexible if and only if it can be continuously deformed while preserving the distance constraints; otherwise the framework is rigid. A framework is redundantly rigid if and only if one can remove any edge and the remaining framework is rigid.*

**Definition 6.7** (Infinitesimal motion). [Hen] *An **infinitesimal motion** is an assignment of velocity to each vertex such that  $(v_i - v_j)(f_i - f_j) = 0$  for all pairs  $(i, j) \in E$ . A framework  $f(G)$  is infinitesimally rigid if and only if it does not have any infinitesimal motion.*

**Theorem 6.8** (Gluck 1975). [Glu] *If a graph has a single infinitesimally rigid realization, then all its generic realizations are rigid.*

The main tool we will use to study the rigidity of our crescent configurations, as mentioned, are **rigidity matrices**. Each framework has a rigidity matrix associated to it. This matrix is the set of all equations whose solutions are the infinitesimal motions of that framework. Its rows correspond to the edges and its  $nd$  columns correspond to the components of the vertices in  $\mathbb{R}^d$ . There are  $2d$  nonzero elements in each row, one for each coordinate of the vertices connected to the corresponding edge. The differences in the coordinate values for the two vertices are these nonzero values.

**Theorem 6.9** (Hendrickson 1992). [Hen] *A framework  $f(G)$  is rigid if and only if its rigidity matrix has rank exactly equal to  $S(n, d)$ , the number of allowed motions, which is equal to  $nd - d(d+1)/2$  for  $n \geq d$  and  $n(n-1)/2$  otherwise.*

All the above theorems and objects are tied together in the following lemma, which provides the basis for our analysis in the next section.

**Lemma 6.10.** [Jac] *The following statements are equivalent:*

- (1)  $G$  is rigid in  $\mathbb{R}^d$ ,
- (2) some framework  $f(G)$  in  $\mathbb{R}^d$  is infinitesimally rigid,
- (3) every generic framework of  $G$  in  $\mathbb{R}^d$  is rigid.

**6.2. Rigidity Analysis for 4-point Crescent Configurations.** The first thing we need to do before we are able to study the rigidity matrix is to construct a realization. By Theorem 6.8, it suffices to construct a single realization for each type of crescent configurations on  $n$  points and study the rigidity of that framework.

Figure 6 is a realization of type C obtained by fixing  $d_1 = 1$ . We note that since the distances in these configurations must satisfy general position as well as geometric realizability, fixing one distance to calculate the rest of the distances in the distance set does not affect the rigidity characterization of the configuration.

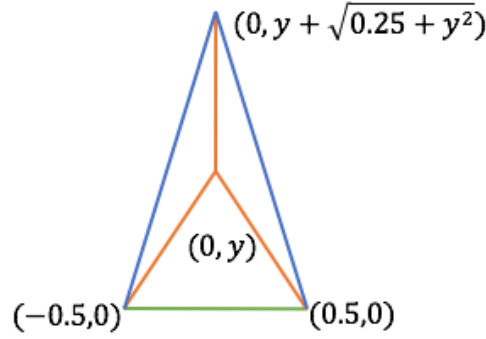


FIGURE 6. Realization of type C obtained by fixing  $d_1 = 1$

The above realization yield the following rigidity matrix, which we denote  $A_C$ :

$$A_C = \begin{bmatrix} \frac{1}{2} & y + \sqrt{\frac{1+4y^2}{4}} & -\frac{1}{2} & -y - \sqrt{\frac{1+4y^2}{4}} & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & y + \sqrt{\frac{1+4y^2}{4}} & 0 & 0 & \frac{1}{2} & -y - \sqrt{\frac{1+4y^2}{4}} & 0 & 0 \\ 0 & \sqrt{\frac{1+4y^2}{4}} & 0 & 0 & 0 & 0 & 0 & -\sqrt{\frac{1+4y^2}{4}} \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & -y & 0 & 0 & \frac{1}{2} & y \\ 0 & 0 & 0 & 0 & \frac{1}{2} & -y & -\frac{1}{2} & y \end{bmatrix}.$$

By row reduction operations, we get that  $\text{Rank}(A_C) = 5 = S(4, 2)$ . Thus by Theorem 6.9, the framework is infinitesimally rigid. Since the row reduction operations yield the same result whether  $y > 0$  or  $y < 0$ , we can conclude that type C is rigid for all  $y \neq 0$  by Lemma 6.10.

Similarly, we can carry out the same analysis on type M. There are two realizations of type M, which are included in Figure 7.

Note that  $d_1$  is the only distance that differs in these two realizations. If we remove  $d_1$  and denote the remaining framework as  $M'_1$  and  $M'_2$  respectively, we could continuously deform one into the other. This fact also means that type M cannot be redundantly rigid by definition. Therefore, type M is another rigid graph.

Last but not least, type R can also be studied using the same method. We start with a realization that is included in Figure 8 and a rigidity matrix  $A_R$  to follow:

$$A_R = \begin{bmatrix} -x & 0 & x & 0 & 0 & 0 & 0 & 0 \\ \frac{-x}{2} & \frac{-x}{2} & 0 & 0 & \frac{x}{2} & \frac{x}{2} & 0 & 0 \\ \frac{-1}{2x} & \frac{-y}{2x} & 0 & 0 & 0 & 0 & \frac{1}{2x} & \frac{y}{2x} \\ 0 & 0 & x - \frac{x}{2} & \frac{-x}{2y} & -x + \frac{x}{2} & \frac{x}{2y} & 0 & 0 \\ 0 & 0 & x - \frac{1}{2x} & \frac{-y}{2x} & 0 & 0 & -x + \frac{1}{2x} & \frac{y}{2x} \\ 0 & 0 & 0 & 0 & \frac{x}{2} - \frac{1}{2x} & \frac{x}{2y} - \frac{y}{2x} & \frac{-x}{2} + \frac{1}{2x} & \frac{-x}{2y} + \frac{y}{2x} \end{bmatrix}.$$

14

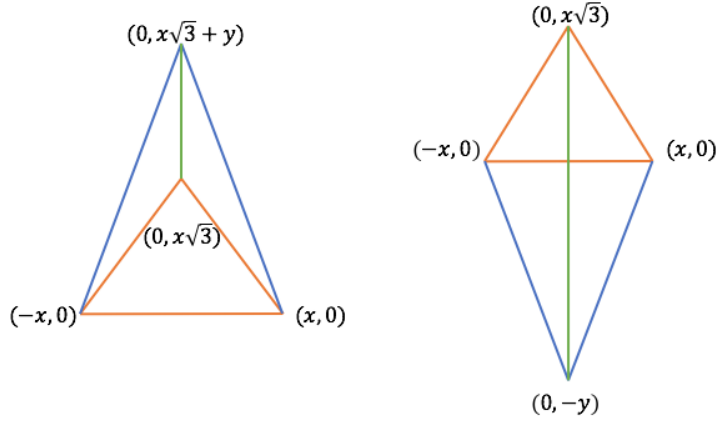


FIGURE 7. Two realizations of type M:  $M_1$  and  $M_2$

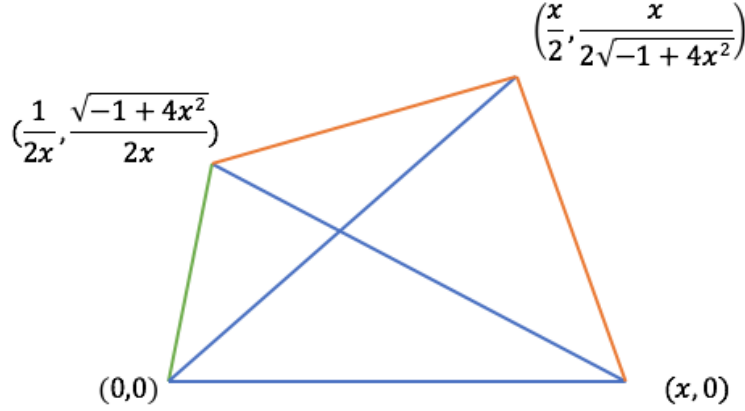


FIGURE 8. Realization of type R obtained by fixing  $d_1 = 1$

By carrying out row reduction operations once more, we find that  $\text{Rank}(A_R) = 6 > S(4, 2)$ . However, since removing any row in the matrix is equivalent to removing any edge in the framework and the rank of any remaining matrix obtained this way is always 5, we conclude that type R is redundantly rigid by Definition 6.6. By [Hen], the conditions for unique realization of a graph is rigidity,  $(d + 1)$ -connectedness, and redundant rigidity. Since type R satisfies all these conditions, it immediately follows that for each value of  $x$  that we choose, there is a *unique realization* of type R.

In conclusion we have found that not all crescent configurations on four points have the same rigidity. As stated above, only type R has a unique realization in  $\mathbb{R}^2$ . Similarly, this analysis may be done on all 27 crescent configurations on 5 points to determine which ones have a unique realization in  $\mathbb{R}^2$ .

As mentioned previously, rigidity matrices are very accommodating. Suppose we need to lift our crescent configurations into  $\mathbb{R}^d$  for  $d > 2$ , which we discuss in Subsection 7.2. Then, the

number of rows and columns of the associated rigidity matrices as well as  $S(n, d)$  would change but the method of evaluating rigidity would remain the same. Similarly, with the extension of our definition from crescent configuration to  $\epsilon$  crescent configuration (7.3), we can insert  $\epsilon$  into the matrix by constructing a realization that involves  $\epsilon$  and then solve for  $\epsilon$  depending on what stability type we want the configuration to exhibit.

## 7. FUTURE WORK

**7.1. Further Explorations in the Plane.** Thus far, we have used our techniques to classify crescent configurations in the plane for  $n = 4$  and  $n = 5$ . Because of the complexity of our algorithm, we have not been able to apply our techniques to higher  $n$ . As mentioned above, the runtime of our current algorithm is on the order of  $n^n$ , which prevents us from carrying out this process for large  $n$ . However, thus far no configurations have been found for  $n > 8$ , so even running a similar algorithm for  $n = 9$  would yield significant progress on this problem. Thus, we are interested in the possibility of modifying our algorithm or finding a new technique that would allow us to count crescent configurations on higher  $n$ . In this way, we could develop a sequence of  $\{c_i\}$ , where each  $c_i$  gives the number of crescent configurations on  $i$  points. If Erdős’ conjecture is correct, then  $\{c_i\}$  only has a finite number of non-zero terms. It would be interesting to see Erdős’ conjecture realized as a sequence that goes to zero.

Since our techniques yield every possible crescent configuration for a given  $n$ , we can use these to observe patterns. For example, one can see from Figure 2 that many of crescent configurations on 5 points contain crescent configurations on 4 points. We may be able to develop techniques using such patterns that generate some of the possible crescent configurations for larger  $n$ .

**7.2. Extensions to Higher Dimensions.** As mentioned earlier, the distance geometry techniques that we use naturally extend to higher dimensions. Thus, we are interested in using these techniques to find the number of crescent configurations on  $n$  points in a given dimension. Our goal is to construct a sequence for each  $d$  consisting of the number of crescent configurations on  $i$  points in  $\mathbb{R}^d$  for each  $i$ . Currently, constructions in  $\mathbb{R}^3$  have been found for 3, 4, and 5 points. Thus, even finding a single 6 point configuration in 3D would give new information. We have attempted to use techniques from distance geometry to find a realization in  $\mathbb{R}^3$  of a known distance set for  $n = 6$  in the plane. However, the resulting systems of equations exceeded our computational resources.

Recently, Burt et. al [BGMMPs] found that given  $d$  high enough, one can always construct a crescent configuration on  $n$  points in  $\mathbb{R}^d$ . We can consider similar questions using the concept of distance coordinates. We are interested in determining whether given a distance set there always exists a dimension in which the set is geometrically realizable.

**7.3. Properties of Crescent Configuration Types.** Now that we have developed a way of classifying crescent configurations, we can examine certain properties for each of the types of crescent configurations. We started to explore in this direction with our rigidity calculations. One direction we are interested in is to develop a concept of stability for these configurations, as we noticed that moving the points of the M, R, and C- type configurations resulted in different amounts of change in the distances. Further, should we define two distances to be equal if they are  $\epsilon$  apart, then our study of the stability of crescent configurations could have some powerful applications to the study of molecules.



## APPENDIX A.

Below we include pseudocode for Algorithms A.1 and A.3.

**Algorithm A.1:** CRESCENTCLASSIFICATION( $distances, n$ )

**procedure** PERMUTATIONS( $distances$ )

**do**  $data \leftarrow$  all permutations of  $distances$   
**return** ( $d$ ) $ata$

**procedure** ADJACENCYMATRIX( $distances, n$ )

**for each**  $w \in$  PERMUTATIONS( $distances$ )  
**do** Thread the list through an  $n \times n$  matrix to create a symmetric matrix with diagonal 0.

**procedure** REORDER( $distances, n$ )

**for each**  $x \in$  ADJACENCYMATRIX( $distances, n$ )  
**do** Sort the elements of each row into canonical order, then sort the rows by first element.

**comment:** Check all 4 – point subsets for isosceles trapezoids

**procedure** SUBMATRICES( $distances, n$ )

**for each**  $y \in$  ADJACENCYMATRIX( $distances, n$ )  
**do**  $k \leftarrow$  Take all  $4 \times 4$  submatrices taken  
along the diagonal.  
**return** ( $k$ )

**procedure** REMOVECYCLIC( $distances, n$ )

**for each**  $z \in$  REORDER( $distances, n$ )  
**do if** No  $i \in$  SUBMATRICES( $distances, n$ ) defines a cyclic quadrilateral or circle  
**then return** ( $z$ )  
**else return** (0)

**procedure** REMOVELINEARCASE( $distances, n$ )

**for each**  $z \in$  REMOVECYCLIC( $distances, n$ )  
**do if**  $\exists$  3 or more isosceles triangles sharing the same base  
**then return** (0)  
**else return** ( $z$ )

**procedure** REMOVEDUPLICATES( $distances, n$ )

**for each**  $i \in$  REMOVELINEARCASE( $distances, n$ )  
**do if**  $i = 0$   
**then** Delete  $i$   
**else** Delete all duplicates of  $i$ , keeping only first instance.

**Algorithm A.2:** CRESCENTCLASSIFICATIONCONTINUED( $distances, n$ )

```

procedure FINALFORM( $distances, n$ )
  for each  $j \in \text{REMOVEDUPLICATES}(distances, n)$ 
    do return (ADJACENCYMATRIX( $distances, n$ )).

```

```

main
   $(x, y) \leftarrow (distances, n)$ 
  output (FINALFORM( $x, y$ ))

```

**Algorithm A.3:** GEOMETRYCHECK( $distances, n$ )

```

procedure SUBMATRICES( $distances, n$ )
  for each  $y \in \text{CRESCENTCLASSIFICATION}(distances, n)$ 
    do  $k \leftarrow$  Take all  $4 \times 4$  submatrices taken
    along the diagonal.
  return ( $k$ )

```

```

procedure CAYLEYMENDER( $distances, n$ )
  for each  $k \in \text{SUBMATRICES}(distances, n)$ 
    do  $S \leftarrow$  Take Cayley – Menger determinants of each  $4 \times 4$  submatrix and set
    equal to zero
  return ( $S$ )

```

**comment:** This generates the system of equations used to solve for  $\{d_2, d_3, \dots, d_n\}$ .

```

procedure SOLUTIONSCHECK( $distances, n$ )
  for each  $S \in \text{CAYLEYMENDER}(distances, n)$ 
    do  $\{d_2, d_3, \dots, d_n\} \leftarrow$  Solution for  $S$ .
  if  $\{d_2, d_3, \dots, d_n\} \subset (0, \infty)$ 
    then return (CRESCENTCLASSIFICATION( $distances, n$ ))
    else return (NULL)

```

```

procedure LINECHECK( $distances, n$ )
  for each  $y \in \text{CAYLEYMENDER}(distances, n)$ 
    do  $y \leftarrow$  Set of Cayley – Menger determinants of all
     $3 \times 3$  submatrices of  $y$  taken along the diagonal
  if 0 in  $y$ 
    then return (NULL)
    else return (SOLUTIONSCHECK( $distances, n$ ))

```

**comment:** By 4.4, the Cayley – Menger determinant of a  $3 \times 3$  matrix will indicate if the three points lie on the same line.

**Algorithm A.4:** GEOMETRYCHECKCONTINUED( $distances, n$ )

**procedure** EUCLIDEANDISTANCEMATRIX( $distances, n$ )

$f \leftarrow x^2$

**do for each**  $y \in \text{LINECHECK}(distances, n)$   $k \leftarrow \text{Map } f \text{ to each element of } y$

**return** ( $k$ )

**comment:** The next procedure is a direct application of Theorem 4.7.

**procedure** CIRCLECHECK( $distances, n$ )

**for each**  $k \in \text{EUCLIDEANDISTANCEMATRIX}(distances, n)$

**do**  $j \leftarrow$  the set of determinants of all  $4 \times 4$  submatrices taken along the diagonal of  $k$

**if**  $0 \in j$

**then return** ( $NULL$ )

**else return** ( $\text{SOLUTIONSCHECK}(distances, n)$ )

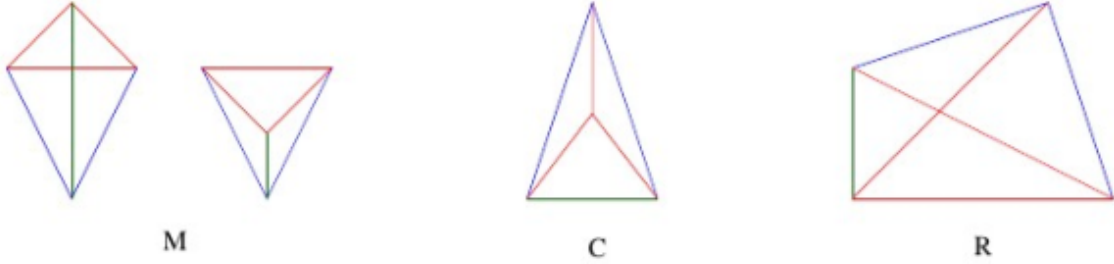
**main**

$(x, y) \leftarrow (distances, n)$  **output** ( $\text{CircleCheck}x, y$ )

APPENDIX B.

Below we provide a list of adjacency matrices and distances for each configuration shown in Figures 1 and 2. These represent members from all possible isomorphism classes of configurations on four and five points. We say  $d_1 = 1$  for all configurations on five points.

Note that most of these solutions are irrational and many of them have no *nice* form (we consider  $\frac{1}{\sqrt{2}}$  and  $\sqrt{1 - \sqrt{3}}$  to be *nice* forms). In these cases, numerical values are provided up to four decimal places. Contact the author for a list of Mathematica outputs.



$$\{d_1, d_2, d_2\}, \{d_1, d_3, d_3\}, \\ \{d_2, d_3, d_3\}, \{d_2, d_3, d_3\}$$

$$\{d_1, d_2, d_3\}, \{d_1, d_2, d_3\}, \\ \{d_2, d_2, d_3\}, \{d_3, d_3, d_3\}$$

$$\{d_1, d_2, d_3\}, \{d_1, d_3, d_3\}, \\ \{d_2, d_3, d_2\}, \{d_3, d_3, d_2\}$$

$$\{d_1 = \sqrt{3} + \sqrt{8}, d_2 = 3, d_3 = 2\} \quad \{d_1 = 1, d_2 = \frac{\sqrt{17}}{2}, d_3 = \frac{17}{16}\} \quad \{d_1 = \sqrt{\frac{3}{2}}, d_2 = 2, d_3 = 1\}$$

FIGURE 9. Types M, C, and R with distance coordinates and values.



Continuation of Table 1														
(22) $\begin{pmatrix} 0 & 1 & d_2 & d_4 & d_4 \\ 1 & 0 & d_3 & d_3 & d_4 \\ d_2 & d_3 & 0 & d_2 & d_4 \\ d_4 & d_3 & d_2 & 0 & d_3 \\ d_4 & d_4 & d_4 & d_3 & 0 \end{pmatrix}$					(23) $\begin{pmatrix} 0 & 1 & d_2 & d_4 & d_4 \\ 1 & 0 & d_4 & d_3 & d_4 \\ d_2 & d_4 & 0 & d_3 & d_3 \\ d_4 & d_3 & d_3 & 0 & d_2 \\ d_4 & d_4 & d_3 & d_2 & 0 \end{pmatrix}$					(24) $\begin{pmatrix} 0 & 1 & d_3 & d_3 & d_4 \\ 1 & 0 & d_3 & d_4 & d_4 \\ d_3 & d_3 & 0 & d_2 & d_2 \\ d_3 & d_4 & d_2 & 0 & d_4 \\ d_4 & d_4 & d_2 & d_4 & 0 \end{pmatrix}$				
(25) $\begin{pmatrix} 0 & 1 & d_3 & d_3 & d_4 \\ 1 & 0 & d_4 & d_3 & d_4 \\ d_3 & d_4 & 0 & d_2 & d_2 \\ d_3 & d_3 & d_2 & 0 & d_4 \\ d_4 & d_4 & d_2 & d_4 & 0 \end{pmatrix}$					(26) $\begin{pmatrix} 0 & 1 & d_4 & d_3 & d_3 \\ 1 & 0 & d_4 & d_3 & d_4 \\ d_4 & d_4 & 0 & d_2 & d_2 \\ d_3 & d_3 & d_2 & 0 & d_4 \\ d_3 & d_4 & d_2 & d_4 & 0 \end{pmatrix}$					(27) $\begin{pmatrix} 0 & 1 & d_2 & d_4 & d_4 \\ 1 & 0 & d_4 & d_3 & d_4 \\ d_2 & d_4 & 0 & d_2 & d_3 \\ d_4 & d_3 & d_2 & 0 & d_3 \\ d_4 & d_4 & d_3 & d_3 & 0 \end{pmatrix}$				

TABLE 2. Realizable distances for each matrix in Table 1.

Realizable Distances for Five Points	
(1)	$\{d_2 \rightarrow \sqrt{\frac{6-3\sqrt{2}-\sqrt{6(3-2\sqrt{2})}}{2(2-\sqrt{2})}}, d_3 \rightarrow \sqrt{\frac{2-\sqrt{2}}{2}}, d_4 \rightarrow \sqrt{\frac{4-2\sqrt{2}-\sqrt{6(3-2\sqrt{2})}}{2}}\}$
(2)	$\{d_2 \rightarrow \sqrt{\frac{4-\sqrt{3}}{2}}, d_3 \rightarrow \sqrt{\frac{2-\sqrt{3}}{2}}, d_4 \rightarrow \frac{1}{\sqrt{2}}\}$
(3)	$\{d_2 \rightarrow \sqrt{\frac{1+\sqrt{3}}{2}}, d_3 \rightarrow \frac{1}{2}(-1 + \sqrt{3+2\sqrt{3}}), d_4 \rightarrow \frac{1}{2}\sqrt{2-2\sqrt{-3+2\sqrt{3}}}\}$
(4)	$\{d_2 \rightarrow 1.2091, d_3 \rightarrow 0.5028, d_4 \rightarrow 0.8135\}$
(5)	$\{d_2 \rightarrow \frac{1}{2}\sqrt{\frac{13+\sqrt{73}}{2}}, d_3 \rightarrow \frac{1}{2}\sqrt{\frac{23+3\sqrt{73}}{2}}, d_4 \rightarrow \frac{1}{2}\sqrt{\frac{9+\sqrt{73}}{2}}\}$
(6)	$\{d_2 \rightarrow \frac{1}{\sqrt{2}}, d_3 \rightarrow \sqrt{\frac{3+\sqrt{6}}{6}}, d_4 \rightarrow \sqrt{\frac{3}{2} + \sqrt{\frac{3}{2}}}\}$
(7)	$\{d_2 \rightarrow \frac{1}{\sqrt{3}}, d_3 \rightarrow \sqrt{\frac{2}{3}}, d_4 \rightarrow \sqrt{1 + \sqrt{\frac{2}{3}}}\}$
(10)	$\{d_2 \rightarrow 0.2757, d_3 \rightarrow 0.5107, d_4 \rightarrow 0.7621\}$
(9)	$\{d_2 \rightarrow \sqrt{2-\sqrt{3}}, d_3 \rightarrow \sqrt{\frac{2-\sqrt{3}}{2}}, d_4 \rightarrow \frac{1}{\sqrt{2}}\}$
(10)	$\{d_2 \rightarrow \sqrt{\frac{2(4-\sqrt{13})}{-1+\sqrt{13}}}, d_3 \rightarrow \sqrt{\frac{4-\sqrt{13}}{3}}, d_4 \rightarrow \sqrt{\frac{-1+\sqrt{13}}{6}}\}$
(11)	$\{d_2 \rightarrow \sqrt{\frac{-35+19\sqrt{13}}{17(9-\sqrt{13})}}, d_3 \rightarrow \sqrt{\frac{9-\sqrt{13}}{34}}, d_4 \rightarrow \sqrt{\frac{9-\sqrt{13}}{34}}\}$
(12)	$\{d_2 \rightarrow \sqrt{\frac{1+\sqrt{3}}{2}}, d_3 \rightarrow \frac{-1+\sqrt{3+2\sqrt{3}}}{2}, d_4 \rightarrow \frac{1}{\sqrt{2+3\frac{1}{4}\sqrt{2}}}\}$
(13)	$\{d_2 \rightarrow 0.3383, d_3 \rightarrow 0.8135, d_4 \rightarrow 0.5028\}$
(14)	$\{d_2 \rightarrow \sqrt{8-3\sqrt{7}}, d_3 \rightarrow \sqrt{\frac{2(45-17\sqrt{7})}{8-3\sqrt{7}}}, d_4 \rightarrow \sqrt{3-\sqrt{7}}\}$
(15)	$\{d_2 \rightarrow 1.9696, d_3 \rightarrow 1.5321, d_4 \rightarrow 2.8794\}$
(16)	$\{d_2 \rightarrow 0.7597, d_3 \rightarrow 1.2293, d_4 \rightarrow 0.5112\}$
(17)	$\{d_2 \rightarrow \sqrt{4+\sqrt{13}}, d_3 \rightarrow \frac{1}{2}(3+\sqrt{13}), d_4 \rightarrow \sqrt{\frac{1}{2}(3+\sqrt{13})}\}$
(18)	$\{d_2 \rightarrow 0.3976, d_3 \rightarrow 0.5304, d_4 \rightarrow 0.7944\}$
(19)	$\{d_2 \rightarrow 1.0879, d_3 \rightarrow 0.5154, d_4 \rightarrow 0.6344\}$

Continuation of Table 2	
(20)	$\{d_2 \rightarrow 1.3275, d_3 \rightarrow 2.0277, d_4 \rightarrow 1.0730\}$
(21)	$\{d_2 \rightarrow 1.1578, d_3 \rightarrow 0.9345, d_4 \rightarrow 1.8686\}$
(22)	$\{d_2 \rightarrow 1.1561, d_3 \rightarrow 0.6707, d_4 \rightarrow 0.5801\}$
(23)	$\{d_2 \rightarrow \sqrt{8 - 3\sqrt{7}}, d_3 \rightarrow \sqrt{\frac{45-17\sqrt{7}}{8-3\sqrt{7}}}, d_4 \rightarrow \sqrt{2\frac{45-17\sqrt{7}}{8-3\sqrt{7}}}\}$
(24)	$\{d_2 \rightarrow 0.3107, d_3 \rightarrow 0.5028, d_4 \rightarrow 0.6180\}$
(25)	$\{d_2 \rightarrow \frac{\sqrt{4-\sqrt{7}}}{3}, d_3 \rightarrow \frac{1}{3}\sqrt{\frac{13-\sqrt{7}}{4-\sqrt{7}}}, d_4 \rightarrow \frac{1}{3}(-1 + \sqrt{7})\}$
(26)	$\{d_2 \rightarrow 0.6599, d_3 \rightarrow 1.3930, d_4 \rightarrow 0.8124\}$
(27)	$\{d_2 \rightarrow \sqrt{2}, d_3 \rightarrow \sqrt{2(3 - \sqrt{7})}, d_4 \rightarrow \sqrt{3 - \sqrt{7}}\}$

## REFERENCES

- [AR] L. Asimow and B. Roth, "The Rigidity of Graphs", *Trans. Amer. Math. Soc.*, **245** (1978), 279–289.
- [Hen] B. Hendrickson, "Conditions for Unique Graph Realizations", *SIAM Journal On Computing*, **21** (1992), 65–84.
- [Jac] B. Jackson, "Notes on Rigidity of Graphs", *Levico Conference Notes*, (2007), 4.
- [Con] R. Connelly, "On generic global rigidity", *Applied Geometry and Discrete Mathematics, the Victor Klee Festschrift, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Vol. 4 (P.Gritzmann and B. Sturmfels, eds.), AMS and ACM, 1991, 147–155.
- [DS] C. D’Andrea, M. Sombra, "The Cayley-Menger determinant is irreducible for all  $n \geq 3$ ", *Siberian Journal of Mathematics*, **46** (2005), 46–71.
- [Erd46] P. Erdős, "On sets of distances of  $n$  points", *Amer. Math. Monthly*, **53** (1946), 248–250.
- [Erd89] P. Erdős, "Distances with specified multiplicities", *Amer. Math. Monthly*, **96** (1989), 447.
- [Erd96] P. Erdős, "On some of my favourite theorems", *Combinatorics, Paul Erdős is Eighty*, Vol. 2 (D. Miklós et al., eds.), Bolyai Society Mathematical Studies 2, Budapest, 1996, 97–132.
- [GK] L. Guth and N.H. Katz, "On the Erdős distinct distances problem in the plane", *Annals of Mathematics*, **181** (2015), 155–190.
- [HHH] S. Hsieh, C. Hsu and L. Hsu, "Efficient Method to Perform Isomorphism Testing of Labelled Graphs", *Computational Science and its Applications– ICCSA 2006: International Conference, Glasgow, UK, May 8-11, 2006, Proceedings*, Part 5 (M. Gavrilova, et al, eds.), Springer, 2006, 422–431.
- [Hal] G. B. Halsted, *Elementary Synthetic Geometry*, J. Wiley and Sons (1896), 50–53.
- [Glu] H. Gluck, "Almost all simply connected closed surfaces are rigid", *Geometric topology*, Springer Berlin Heidelberg (1975), 225–239.
- [Lam] G. Laman, "On graphs and rigidity of plane skeletal structures", *J. Eng. Math.*, **4** (1970), 331–340.
- [LL] L. Liberti, C. Lavor, "Six mathematical gems from the history of distance geometry.", *International Transactions in Operational Research*, **23** (2016), 897–920.
- [Pal87] I. Palásti, "On the seven points problem of P. Erdős", *Stud. Sci. Math. Hungar.*, **22** (1987), 447–448.
- [Pal89] I. Palásti, "Lattice point examples for a question of Erdős", *Period. Math. Hungar.*, **20** (1989), 231–235.
- [Rot80] B. Roth, "Questions on the rigidity of structures", *Topologie Structurale*, **4** (1980), 67–71.
- [Rot81] B. Roth, "Rigid and flexible frameworks", *Amer. Math. Monthly*, **88** (1981), 6–21.
- [She] A. Sheffer, "Distinct Distances: Open Problems and Current Bounds", (2015). arXiv:1406.1949.
- [BGMMPs] D. Burt, E. Goldstein, S. Manksi, S.J. Miller, E.A. Palsson and H. Suh, "Crescent Configurations", *Integers*, **16** (2016), #A38. arXiv:1509.07220.

*E-mail address:* [rfd1@williams.edu](mailto:rfd1@williams.edu)

DEPARTMENT OF MATHEMATICS AND STATISTICS, WILLIAMS COLLEGE, WILLIAMSTOWN, MA 01267

*E-mail address:* [mhlavacek@g.hmc.edu](mailto:mhlavacek@g.hmc.edu)

DEPARTMENT OF MATHEMATICS, HARVEY MUDD COLLEGE

*E-mail address:* [huynhngocyenchi@gmail.com](mailto:huynhngocyenchi@gmail.com), [nhuynh30@gatech.edu](mailto:nhuynh30@gatech.edu)

SCHOOL OF MATHEMATICS, GEORGIA INSTITUTE OF TECHNOLOGY, ATLANTA, GA 30332

*E-mail address:* [sjm1@williams.edu](mailto:sjm1@williams.edu), [Steven.Miller.MC.96@aya.yale.edu](mailto:Steven.Miller.MC.96@aya.yale.edu)

DEPARTMENT OF MATHEMATICS AND STATISTICS, WILLIAMS COLLEGE, WILLIAMSTOWN, MA 01267

*E-mail address:* [palsson@vt.edu](mailto:palsson@vt.edu)

DEPARTMENT OF MATHEMATICS, VIRGINIA TECH UNIVERSITY, BLACKSBURG, VA 24061